Option Adjusted Model (OAS) Model
Pricing Callable Bonds

Supervisor:
Jan R. M. Röman

Authors:
Natalia Arango Mesa
Junior Gutierrez Berrones
Danuwat Weereerat

Västerås - 2007
This work introduce the Option Adjusted Spread (OAS) as an instrument to value zero coupon bonds and bonds whit embedded options, in order to accomplish it we first present in a simple way the concepts used, and in a more detailed way, the theories which support the OAS model, those theories include as a central background the Black-Derman-Toy model. Then we continue with an illustrative example finishing with the full explanation of the developed application.
# Contents

1. Theoretical Background ........................................................................................................... 4  
   Definitions ................................................................................................................................. 4  
   Option Adjusted Spread ............................................................................................................ 7  
   Building the OAS ....................................................................................................................... 7  

2. Example ...................................................................................................................................... 11  
   Calibrating the tree .................................................................................................................... 11  
   Finding the spread ...................................................................................................................... 11  
   Calculating the bond price ......................................................................................................... 12  
   Calculating the value for the embedded option ......................................................................... 12  

3. VBA Application ....................................................................................................................... 13  
   Structure ................................................................................................................................... 13  
   Output ....................................................................................................................................... 14  
   Analysis ..................................................................................................................................... 16  

4. Conclusion .................................................................................................................................. 18  

5. Bibliography ............................................................................................................................. 19  

6. Appendix ................................................................................................................................... 19
1. Theoretical Background

First it is necessary to describe the most important concepts.

**Definitions**

**Bond yield** is the discount rate that when applied to all cash flows of a bond, gives a price equal to its market price.

**Yield to Maturity** is the interest rate which gives you the market price of a bond. Therefore we can find the YTM by solving the following equation

\[ P = \sum_{i} \frac{c_i}{(1 + YTM)^{t_i}} + \frac{100}{(1 + YTM)^e} \]

Where P is the market price of the bond, \( c_i \) are the coupons of the bond and \( t_i \) the time for the payouts. With continuous interest rate (continuous compounding) we can write this as

\[ P = \sum_{i} c_i \cdot e^{-t_i \cdot YTM} + 100 \cdot e^{-e \cdot YTM} \]

We have to reinvest all the coupons at the same yield and that the YTM is unique for each instrument. If we have no possibility to reinvest the coupons at the same yield we replace the term \((1 + YTM)^e\) above with \((1 + (t)YTM)\).

We have to know the exact time (day) for each coupon payment to get the correct price. Therefore, we have to use the correct day count convention. We therefore introduce the following time periods

\[ t_i = f + (i - 1) \]
\[ t_n = f + (n - 1) + g \]

Where

\[ f = \text{part of period to the next coupon} \]
\[ g = \text{part of period from the last coupon} \]
\[ n = \text{number of whole periods to coupon number } i \]

If the day of maturity doesn’t coincide with a coupon day we have to add a part of a coupon of the size \( \frac{C}{H} \). \( C \) is the coupon rate, \( H \) the number of coupons per year.

**Bullet bond** is a conventional bond paying a fixed periodic coupon and having no embedded option. Such bonds are non-amortizing, i.e., the principal remains the same throughout the life of the bond and is repaid in its entirety at maturity. Bullet bonds are also called *straight bonds*. In the United States such bonds usually pay a semi-annual coupon. The coupon rate (CR) is stated as an annual rate (usually with semi-annual compounding) and paid on the principal amount on the bond or par value (Par). Thus, a single coupon payment is equal to:
**Benchmark bullet bond** is a bullet bond issued by the sovereign government and assumed to have no credit risk (e.g., Treasury bond).

**Non-benchmark bullet bond** is a bullet bond issued by an entity other than the sovereign and which, therefore, has some credit risk.

**Callable bond** is a bond for which the issuer has the right, but not the obligation, to call back/repurchase the bond at one or more specified points over the bond's life. If called, the issuer pays the investor the pre-specified call price, the strike. *The call price is usually higher than the bond's par value.* For the investor, this means that there is uncertainty as to the true maturity of the loan.

\[
\text{Callable bond} = \text{Bullet bond} - \text{Callable bond option price}
\]

**Call premium** is the difference between the call price and the par value of a callable bond.

\[
\text{Call premium} = \text{Call price} - \text{Par}
\]

**Term structure of interest rate** is a time structure that describes the dynamics of the market interest rates. I.e., the relation between the rates on e.g., treasury bonds with different times to maturity. This is described as a graph, the *yield curve*. This yield curve is used to discount cash flows to a present value.

![Graph showing yield curve](image)

**The Black-Derman-Toy (BDT) model** is a lognormal model that is able to capture a realistic term structure of the interest rate volatilities. And also the short rate volatility is allowed to vary over time, and the drift in interest rate movements depends on the level of rates.

Is a single-factor short-rate model matching the observed term structure of forward rate volatilities, as well as the term structure (yield curve) of the interest rate.

The model is described by a stochastic differential equation given by

\[
dr = \alpha(t) dt + \sigma(t) dz
\]

Where \(z(t)\) is a Brownian motion. In some literature this SDE is written as
\[
d\ln(r) = \left( \theta(t) + \rho(t) \ln(r) \right) dt + \sigma(t) dz
\]

Where \( \theta(t) \) will be shown to be the drift of the short-term rate and \( \rho(t) \) the mean reversing term to an equilibrium short-term rate that depends on the interest rate local volatility as follows

\[
\rho(t) = \frac{d}{dt} \ln[\sigma(t)] = \frac{\dot{\sigma}(t)}{\sigma(t)}
\]

Where \( \dot{\sigma}(t) = \frac{d}{dt} \ln \sigma(t) \). And by substituting \( \rho(t) \)

\[
d\ln(r) = \left( \theta(t) + \frac{\dot{\sigma}(t)}{\sigma(t)} \ln(r) \right) dt + \sigma(t) dz
\]

since volatility is time dependent, \( \theta(t) \) and \( \sigma(t) \) are two independent functions of time, chosen so that so can have a model that fits the term structure of spot interest rates and the structure of the spot rate volatilities.

The level (in the yield curve) of the short rate at time \( t \) in the Black-Derman-Toy model is given by

\[
r(t) = U(t) \exp \left( \sigma(t) z(t) \right)
\]

Where \( U(t) \) is the median of the lognormal distribution of \( r \) at time \( t \), \( \sigma(t) \) the level of the short rate volatility, and \( z(t) \) is the level of the Brownian motion, a normal distributed Wiener process that captures the randomness of future changes in the short-term rate. Given the properties of the Brownian motion we have

\[
z(t) = \varepsilon \cdot \sqrt{t}
\]

Where the values

\[
\varepsilon = \begin{cases} 
+1 \\
-1 
\end{cases}
\]

are used to build the binomial tree for the OAS, a fixed spaced \( Z_i \) between the nodes of the tree is defined as \( (\varepsilon_{\text{max}} - \varepsilon_{\text{min}} = 2) \)

\[
Z_i = e^{2 \sigma_i \sqrt{t_i - t} - i - 1}
\]

where \( \sigma_i \) is the volatility at time \( t \). The risk-neutral probabilities of the binomial branches are assumed to be a symmetric random walk, this means that the actual probability for increase or decrease of the interest rate are equal to \( \frac{1}{2} \).
The tree uses the short rate annual volatility, \( \sigma \), of the benchmark rates which should be given in the Black-Scholes framework.

**Option Adjusted Spread**

The Option adjusted spread (OAS) is a method used to value zero bonds, bonds and promissory loans with embedded options. To do it the OAS use the Term structure (Yield curve) of the risk free interest rate instrument as a benchmark, and by adding a spread to this benchmark the OAS adjust it to the term structure of the market price of the bond. The spread added is called option-adjusted spread since the context of OAS started with trying to correct for mispricing in option embedded securities.

The value of the spread OAS allow us to compare fixed income instruments with the same characteristics, but traded at very different yields because of embedded options.

To do it a binomial tree is constructed for the short rate in such a way that the tree automatically returns the observed yield function and the volatility of different yields. Then we adjust the theoretical price to the actual price by adding the spread to all short rates on the tree. This new model price after adding this spread makes the model price equal the market price.

The OAS model has three dependent variables; the Option Adjusted Spread, the volatility and the underlying price. And also can calculate the Effective duration, the effective modified duration, effective convexity and the option adjusted spread.

**Building the OAS**

We can summarize the building process of OAS in the follow steps:

1. A binomial benchmark tree with equal for probabilities for up and down of \( \frac{1}{2} \) is built, with their nodes on dates with the following events
   - Cash flows
   - Single call or put events
   - Start or end of call or put periods, no intermediate nodes are created, since there are no dynamical changes between the nodes. For better accuracy in the interval where the bond can be called or putted we can build intermediate nodes in other (all) parts of the tree.

2. Calibrate the benchmark tree to market data by adjusting the nodes until the tree can replicate any cash flow as the discount function given by the benchmark yield curve.

3. Calibrate the model by adding the spread factor to all rates in the benchmark tree, until the model replicate the actual market price (if this price is known) of the callable bond. The result is the bond’s OAS.

4. Apply the same OAS to value a bullet bond with terms identical to the callable/putable bond.

5. Take the difference of the value of the callable bond minus the value obtained for the bullet bond. This difference is the value of the embedded option.
In detail:

**Construction of the benchmark tree**

Using the embedded Brownian motion property of the Black-Derman-Toy model we can define the space between nodes, which is given by

\[ Z_i = e^{2\sigma_i \sqrt{t_i - t_{i-1}}} \]

where \( Z_i \) is known as the spread factor.

We assume for simplicity that annual volatility of the short rate is constant. When the tree is built, the volatility spread \( Z_i \) is kept constant and the tree is build with the following relation between nodes

\[ f_{i,j} = Z_{i}^{j-1} \cdot f_{i,1} \]

where \( f_{1,1} = f_1 \). This result in the following tree:

![Tree Diagram]

where the rate is given by:

\[
\begin{align*}
    f_{2,2} &= Z_1 \cdot f_{1,1} \\
    \frac{1}{2} f_{2,1} + \frac{1}{2} f_{2,2} &= f_2 \\
    \Rightarrow f_{2,1} &= \frac{2 \cdot f_2}{1 + Z_2} \\
    \Rightarrow f_{2,2} &= f_2
\end{align*}
\]

\[
\begin{align*}
    f_{3,3} &= Z_2 \cdot f_{2,1} \\
    f_{3,2} &= Z_3 \cdot f_{3,3} \\
    \Rightarrow f_{3,3} &= \frac{4 \cdot f_3}{1 + 2 \cdot Z_3 + Z_3^2} \\
    \Rightarrow f_{3,2} &= f_3
\end{align*}
\]

and so on. Generally the rates are expressed as:

\[ f_{n+1} = \sum_{i=0}^{n} \binom{n}{i} \cdot Z_i^i \cdot f_{n,i} \Rightarrow f_{n,1} = f_{n,2} = \ldots = f_{n,n} \]

where the rates are lognormally distributed. Therefore, the interest rates cannot be negative.

**Calibration of the binomial benchmark tree**

The calibration process raise or lower the estimates of the rates in the tree so that the value for the cash flows given in the tree exactly equals the value given by the discount function. As this is done, the relationship
\[ f_{i,t} = Z_{i}^{l-1} \cdot f_{i,1} \]

between different nodes must be preserved. First the nodes are calibrated at time 1, then at
time 2, then time 3 and so on. At time 1 the following must hold:

\[
\left( \frac{1}{2} \left( 1 + f_{2,1} \cdot (t_{2} - t_{1}) \right) + \frac{1}{2} \left( 1 + Z_{2} \cdot f_{2,1} \cdot (t_{2} - t_{1}) \right) \right) \cdot \frac{1}{1 + f_{1,1} \cdot (t_{1} - t_{0})} = P(t_{0}, t_{2})
\]

defining the left side of the equation is the price of a cash flow equal 1 given by the tree, and the right
side is the price of the same cash flow given by the discount function \( P(t,T) \), this function
discount any value from \( t=t_{2} \) to \( t=t_{0} \), where \( t_{0} \) is the valuation time, this equation is solved
numerically by a Van Winjgaard-Decker-Brent method. In the equation above, the following
relationship is used:

\[ f_{2,2} = Z_{2} \cdot f_{2,1} \]

Therefore \( f_{2,2} \) can be calculated as soon as \( f_{2,1} \) is known. At the next level, the following
equation needs to be solved (note, it is not necessary to know the size of the cash flow).

\[
\frac{1}{2} \left( \frac{1}{1 + Z_{3} \cdot f_{3,1} \cdot (t_{3} - t_{2})} + \frac{1}{1 + f_{2,2} \cdot (t_{2} - t_{1})} \right) \cdot \frac{1}{1 + f_{1,2} \cdot (t_{1} - t_{0})} + \\
\left( \frac{1}{1 + Z_{3} \cdot f_{3,1} \cdot (t_{3} - t_{2})} + \frac{1}{1 + f_{3,1} \cdot (t_{3} - t_{2})} \right) \cdot \frac{1}{1 + f_{1,3} \cdot (t_{3} - t_{0})} + \\
\left( \frac{1}{1 + f_{2,1} \cdot (t_{2} - t_{1})} + \frac{1}{1 + f_{1,2} \cdot (t_{1} - t_{0})} \right) \cdot \frac{1}{1 + f_{1,3} \cdot (t_{3} - t_{0})} = P(t_{0}, t_{3})
\]

Solving this equation for \( f_{3,1} \) also gives \( f_{3,2} \) and \( f_{3,3} \) from the relations \( f_{3,2} = Z_{3} \cdot f_{3,1} \) and
\( f_{3,3} = Z_{3} \cdot f_{3,2} \).

Using the same method for cash flows, at all times in the tree, the tree will be fully calibrated
to produce the same value as the forward rates. The reason for the previous calibration is
shown in the figure below where the error is caused by the bond’s convexity.
The curvature represents *convexity*. The value of the cash flow, labeled the "calculated value" above, is an average of the two values $V_1$ and $V_2$. Note that this average is higher than the "actual value". After the calibration, the situation is described in the figure below.

So we can get the actual value, in the figure example $6.439\% + 0.003$ and $7.964\% + 0.003$

*Construction and calibration of the tree with the spread (non benchmark)*

For simplicity we assume no transaction cost.

Now consider the calibrated benchmark tree adapted to value a non benchmark (corporate) callable bond.

The general pricing formula at zero spread paying cash flows $C_1, C_2, \ldots, C_n$ at time $T_1, T_2, \ldots, T_n$ is given by

$$\Pi(0,0) = \sum_{i=1}^{n} C_i \left[ \prod_{j=1}^{i} \frac{1}{(1+f_j)^{T_j-T_{j-1}}} \right]$$

With a shift $s$ ($s \neq 0$) in the rate $f_j$, the price is given as:

$$\Pi(s,0) = \sum_{i=1}^{n} C_i \left[ \prod_{j=1}^{i} \frac{1}{(1+f_j+s)^{T_j-T_{j-1}}} \right]$$

If the market price $\Pi$ is given, the formula can be used with different spreads still the spread that equals the market price is found. This spread is called the *implied spread*, so this same spread shall be applied at all nodes.
2. Example

The follow example shows how to price a bond using backward induction to calibrate the BDT binomial tree to the current term structure of zero-coupon yield and zero-coupon volatilities.

Consider a 36-month corporate bond paying an annual coupon 10% of nominal amount. The bond is callable in 2 year (period =2) at 105$. The benchmark term structure of zero-coupon rates and volatilities is shown in the table below. From the rates and volatilities, we will calibrate the Black-Derman-Toy interest rate to the market and price the callable bond

<table>
<thead>
<tr>
<th>Years to Maturity</th>
<th>Zero-Coupon Rates</th>
<th>Zero-Coupon Volatilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.0%</td>
<td>24%</td>
</tr>
<tr>
<td>2</td>
<td>9.5%</td>
<td>22%</td>
</tr>
<tr>
<td>3</td>
<td>10.0%</td>
<td>20%</td>
</tr>
</tbody>
</table>

The bond’s offer price \( t_0 \) is $98.00

**Calibrating the tree**

Build a binomial tree with equal probabilities (1/2) on dates where we have cash flows and also create nodes in the tree where the bond is callable or putable. Calibrate the tree to market data by adjusting the nodes until the tree can replicate any cash flow as the discount factor given by the benchmark yield curve.

\[
f_{2,2} = Z_2 \cdot f_{2,1}
\]

For example, \( f_{2,1} \) can be obtained by the following formula

\[
\left( \frac{1/2}{1+f_{2,1}} \right) + \left( \frac{1/2}{1+Z \cdot f_{2,1}} \right) = \frac{1}{1+9\%} = p(0,2)
\]

\( p(0,2) \) is the discount factor given from the yield curve. and so on to the next nodes.

**Finding the spread**

Calibrate the model by adding the spread to all rates in the tree until the model replicate the actual market price (Market price is known) of callable/putable bond.

Since we know the market price of bond, 98$, we again can use a trial and error to find the spread, since we know the market price \( t_0 \) of the bond, $98.00
We have the following tree of the rates + spread:

```
               15.89%
             /      \
   13.10%     9.88%
        /    \
  10.89%  8.75%
         /  \
7.89%
```

With trial and error, the spread (s) is adding to all rates until the model replicates the market price, 0.88%

```
110.0000
111.8001
111.3242
  98.0000
  104.0405
    104.1393
      110.0000
```

**Calculating the bond price**
Apply the same OAS to value bullet bond with terms identical to the callable/putable bond.

Using the same OAS, we compute the bullet bond as the result above.

\[ B_{\text{bullet}} = 98.00 \]

**Calculating the value for the embedded option**
Take the difference between the value calculated for the callable/putable bond and the value calculated for the bullet bond. We obtain the value of the embedded option.

The bond is callable in period 2 at 105$.

```
110.0000
105.0000
106.5500
  94.4019
  102.4575
    104.1393
      110.0000
```

\[ B_{\text{callable}} = B_{\text{bullet}} - C_{\text{bullet}} \]

\[ 94.4019 = 98.0000 - C_{\text{bullet}} \]
3. VBA Application

Structure

The application has the following structure to obtain the input data

<table>
<thead>
<tr>
<th>Convention</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annual coupon rate</td>
<td>5.00%</td>
<td></td>
</tr>
<tr>
<td>Coupons per year</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Bond maturity (years)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Steps to maturity</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Times of cash flow</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Spread</th>
<th>Market Bond Price</th>
<th>Bullet Bond Price</th>
<th>Call Option Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00%</td>
<td></td>
<td>96.37</td>
<td>1.27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time (year)</th>
<th>Cash flow</th>
<th>Yield (%)</th>
<th>Volatility (%)</th>
<th>Strike price</th>
<th>Callable period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>5.00</td>
<td>5.00%</td>
<td>29.00%</td>
<td>101.00</td>
<td>Not callable</td>
</tr>
<tr>
<td>2.00</td>
<td>105.00</td>
<td>6.00%</td>
<td>19.00%</td>
<td></td>
<td>Not callable</td>
</tr>
</tbody>
</table>

The calculations executed by the application are as follows

This option constructs, based on the input specifications (such Annual coupon rate, coupons per year and bond maturity) the table in which the user can observe all the cash flow periods and determine for each the Yield, Volatility, Strike price and Callable period (see details below).

Once the previous data is defined, through this option the user can easily build the following binomial trees: Rate tree, the Rate tree using the spread, the Discount factor tree using the spread, Zero coupon bond price using the spread, Coupon paying bond price using the spread and the Callable coupon paying bond price using the spread. Additionally, the Bullet Bond Price and the Call Option Price are calculated and printed on the structure.

Using this option is possible to calculate the Bullet Bond Price and the Call Option Price without constructing the trees. This option can be preferred when higher numbers of steps are used in order to minimize the processing time.

This function generates the option value and the price of bullet bond at different number of step. It compares the value from the tree having the number of step as many as cash flows to the number of step 64 times of the initial nodes. To see the convergence, it also plot graph between those two values and the times of the initial nodes.

When the implied spread is known, this function can be used to calculate the non benchmark market price of bond which have the same OAS.
When the benchmark bond price is known, this function is used to find the spread by adding the same number of the spread factor to all rates in the tree until the model replicate the benchmark bond price (the actual market price).

In this column the user has to specify the condition for the callable period, indicating if it is “Period” or “Spot”. “Period” means that the bond is callable during the selected cashflow and during all intermediate steps until the next cashflow and selecting “Spot” the bond is only callable at the selected cashflow.

Considering that the bond is not callable at maturity, is not possible for the user to modify this field for the last cashflow and this is indicated by a error message.

**Output**

In order to explain how the application works we define the following example

**Input data**

<table>
<thead>
<tr>
<th>Annual coupon rate</th>
<th>Spread</th>
<th>1.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coupons per year</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Bond maturity (years)</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Steps to maturity</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Times of cash flow</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

**Rate tree**

Calibrated interest rates (Aij)

<table>
<thead>
<tr>
<th>Time (year)</th>
<th>Cash flow</th>
<th>Yield</th>
<th>Volatility</th>
<th>Strike price</th>
<th>Callable period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>5.00</td>
<td>5.00%</td>
<td>20.00%</td>
<td>101.00</td>
<td>Period</td>
</tr>
<tr>
<td>2.00</td>
<td>105.00</td>
<td>6.00%</td>
<td>19.00%</td>
<td>Not callable</td>
<td>Not callable</td>
</tr>
</tbody>
</table>

Calibrated interest rates adding the spread (Bij)

<table>
<thead>
<tr>
<th>Time increment (dt)</th>
<th>Coupon rate (cr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dt = Maturity / Steps</td>
<td>cr = Anual coupon / Coupon per year</td>
</tr>
</tbody>
</table>

**Rate tree using the spread**

Calibrated interest rates adding the spread (Bij)

<table>
<thead>
<tr>
<th>Bij = Aij + Spread</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.816%</td>
</tr>
<tr>
<td>5.778%</td>
</tr>
<tr>
<td>5.195%</td>
</tr>
<tr>
<td>6.000%</td>
</tr>
<tr>
<td>6.566%</td>
</tr>
<tr>
<td>9.293%</td>
</tr>
<tr>
<td>11.785%</td>
</tr>
</tbody>
</table>
Discount factor tree using the spread

\[
\begin{array}{c|c|c}
 & 0.9719 & 0.9648 \\
\hline
0.9709 & 0.9648 & 0.9558 \\
0.9682 & 0.9556 & 0.9444 \\
\end{array}
\]

Discount factor \( (C_{ij}) \)

\[
C_{ij} = \frac{1}{1 + (B_{ij} \times dt)}
\]

Zero coupon bond price using the spread

\[
\begin{array}{c|c|c|c|c}
 & 100.0000 & & & \\
\hline
 & 97.1740 & 94.1075 & 87.2915 & \\
\hline
 & 94.4356 & 90.7902 & 88.8052 & \\
\end{array}
\]

Backward calculation starting from bond nominal value (100), and

\[
Node_{ij} = 0.5 \times C_{ij} \times (Node_{i+1, j+1} + Node_{i+1, j} - 1)
\]

Coupon paying bond price using the spread

\[
\begin{array}{c|c|c|c|c|c|c}
 & 105.0000 & & & & & \\
\hline
 & 102.0327 & 103.8129 & 106.4394 & & & \\
\hline
 & 101.5200 & 102.8355 & 103.8129 & & & \\
\hline
 & 99.1574 & 100.3616 & 100.3616 & & & \\
\end{array}
\]

Bullet bond price

Backward calculation starting from nominal value (100) + coupon(1), and

\[
Node_{ij} = 0.5 \times C_{ij} \times (Node_{i+1, j+1} + Node_{i+1, j} - 1) + cr
\]

Callable coupon paying bond price using the spread

\[
\begin{array}{c|c|c|c|c|c|c|c}
 & 105.0000 & & & & & & \\
\hline
 & 101.0000 & 101.0000 & 101.0000 & & & & \\
\hline
 & 101.0000 & 100.3616 & 100.3616 & & & & \\
\hline
 & 99.1574 & 100.3616 & 100.3616 & & & & \\
\end{array}
\]

Callable bond price

Callable bond price = Bullet bond price - Callable bond price

Call option value

\[
1.2707
\]
Analysis
Using the described model it is analyzed how the result behaves when the number of the steps is increased.

Consider a 48-month corporate bond paying an annual coupon 10% of nominal amount. The bond is callable during year 2 to year 3 at $105. The benchmark term structure of zero-coupon rates and other input values are shown in the picture below.

![Image of input and output values]

The following table shows the results for the simulated steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Bullet Bond Price</th>
<th>Option Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90.75479</td>
<td>3.275283</td>
</tr>
<tr>
<td>2</td>
<td>90.20159</td>
<td>3.12016</td>
</tr>
<tr>
<td>4</td>
<td>89.91415</td>
<td>3.029628</td>
</tr>
<tr>
<td>8</td>
<td>89.76761</td>
<td>2.978379</td>
</tr>
<tr>
<td>16</td>
<td>89.69362</td>
<td>2.948392</td>
</tr>
<tr>
<td>32</td>
<td>89.65637</td>
<td>2.937953</td>
</tr>
<tr>
<td>64</td>
<td>89.63772</td>
<td>2.931374</td>
</tr>
</tbody>
</table>

The results are also shown in the graph below

![Graph showing convergence of option value]

The following table shows the results for the simulated steps
Based on the previous information given by the application it is clear that the option value converge to finite value when the times of cash flows increase but the prices do not present any significant variation. The reason of such behavior for the option price is the fact that no coupon is added during intermediate steps (between cash flows) which eliminates the probability of reaching the strike at those points. But the price of option can be different between narrow interval since when we build the finer tree, we also have the accurate ratio of the nodes which above the strike price and the nodes below the strike price.

For the bullet bond price the result is similar to the option price. It can be explained through the fact that we use interpolation between known interest rates to get unknown rates for the new nodes which they are between the two existing nodes. In additional, consider the tree added nodes between two cash flows period, the following tree show that present value of new nodes between two cash flows, not added coupon, equal to the discount from the previous nodes. Thus, the effect of calculation is small since coupon is not added to the further nodes.
4. Conclusion

The tree for the bond and option price calculation should be constructed containing as many nodes as events the bond has, it means that in order to obtain accurate results, all the events for the bond -as cash flows, callable and putable periods- should be represented with nodes in the tree.

Based on the convergence of the pricing value it can be conclude that it is not necessary to build the tree using high number of steps if they do not correspond to bond events.

The developed application calculates the BDT model and calibrates it to both yield and volatility. Using this BDT model the application can calculate the market price of a bond that includes coupon rate, embedded option or spread. If the market price and bond structure is known the applet can also calculate the spread applied to the bond and the results are shown graphically as binomial trees as well as input into the corresponding cells.
5. Bibliography


6. Appendix.

VBA Code

Global f() As Double
Global sigma() As Double
Global Z() As Double
Global Pvbond() As Double
Global fs() As Double
Sub Structure()
    Call ClearFields
    Dim Arate As Double
    Dim Frequence As Long
    Dim Crate As Double
    Dim Nostep As Long
    Dim Maturity As Double
    Worksheets("Variables").Activate
    Arate = Cells(9, 2) 'Annual coupon rate
    Frequence = Cells(10, 2) 'Coupons per year
    Maturity = Cells(11, 2) 'Bond maturity(years)
    Nostep = Cells(12, 2) 'steps to maturity
    Crate = Arate / Frequence
    Cashflow = Frequence * Maturity 'Number of cash flow
    'Time column
    h = Maturity / Cashflow
    For i = 1 To Cashflow
        Cells(15 + i, 1) = h * i
        Next i
    'Cash flow
    For i = 1 To Cashflow - 1
        Cells(15 + i, 2) = Crate * 100
    Next i
    Cells(15 + Cashflow, 2) = (Crate + 1) * 100
    '####################################################
    'Format
    Range("A16", Cells(15 + Cashflow, 2)).Select
    Selection.Style = "Output"
    Selection.NumberFormat = "0.00"
    Range("C16", Cells(15 + Cashflow - 1, 5)).Select
    Selection.Style = "Input"
    Selection.NumberFormat = "0.00%"
    Range("E16", Cells(15 + Cashflow - 1, 6)).Select
    Selection.Style = "Input"
    Range.NumberFormat = "0.00"
    'Selection between Period-Spot for the callable period
    Range("F16", Cells(15 + Cashflow - 1, 6)).Value = "Period"
    With Selection.Validation
        .Delete
        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
        xlBetween, Formula1:="=$A$1:$A$2"
        .IgnoreBlank = False
        .InCellDropdown = True
    End With
"
Callable period specification

Period: The bond is callable during the selected cashflow and during all intermediate steps until the next cashflow.

Spot: The bond is only callable at the selected cashflow.

Remark: The bond can not be callable at maturity.

Range(Cells(15 + Cashflow, 5), Cells(15 + Cashflow, 6)).Value = "Not callable"
Range(Cells(15 + Cashflow, 5), Cells(15 + Cashflow, 6)).Select
Selection.Style = "Output"
With Selection.Validation
.Delete
.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:=xlBetween, Formula1:="=$A$3"
.IgnoreBlank = False
.InCellDropdown = False
.InputTitle = """'
.ErrorMessage = "The callable period must be specified as 'Spot' or 'Period'"
.ShowInput = True
.ShowError = True
End With
Range("G1:ZZ100").Select
Selection.ClearContents
Selection.Style = "Normal"
End Sub
Sub ClearFields()
Rows("16:100").Select
Selection.ClearContents
Selection.Style = "Normal"
End Sub
Sub OAS()
Worksheets("Variables").Activate
Dim Nostep As Integer
Nostep = Cells(12, 2) 'Steps to maturity
Dim Maturity As Double
Maturity = Cells(11, 2) 'Bond Maturity(years)
dt = Maturity / Nostep
Dim epsilon As Double 'Minimum value in Newton-Raphson iterations
epsilon = 0.000001

Dim Error As Double 'Error in Newton-Raphson iterations

Dim h As Double
h = 0.0000001 'h for calculate derivative

Dim Frequency As Integer
Frequency = Cells(10, 2) 'Coupons per year

Dim Cashflow As Integer
Cashflow = Frequency * Maturity 'Number of coupons

t = Nostep / Cashflow

ReDim Yield(Nostep) As Double

If t = 1 Then 'Number of steps equal to cashflow (number of coupons)
    For a = 1 To Cashflow
        Yield(a) = Cells(a + 15, 3)
    Next a
Else 'Linear interpolation for each step
    For a = 1 To Cashflow
        Yield(a * t) = Cells(a + 15, 3)
    Next a
    For a = 1 To t - 1
        Yield(a * t) = Interpolate((Maturity / Cashflow), (2 * Maturity / Cashflow), Yield(t), Yield(2 * t), [a * (Maturity / Nostep)])
    Next a
    For a = 2 To Cashflow
        For b = 1 To t - 1
            Yield(a * t - (t - b)) = Interpolate([a - 1] * Maturity / Cashflow), [a] * Maturity / Cashflow), Yield((a - 1) * t), Yield(a * t), ([[a * t] - (t - b)] * Maturity / Nostep))
        Next b
    Next a
Else 'Extrapolation
    For a = 1 To Cashflow
        sigma(a) = Cells(a + 15, 4)
    Next a
    Else 'Else Interpolate each step
    For a = 1 To Cashflow
        sigma(a * t) = Cells(a + 15, 4)
    Next a
    For a = 1 To t - 1
        sigma(a * t - (t - b)) = Interpolate([a - 1] * Maturity / Cashflow), [a] * Maturity / Cashflow), sigma((a - 1) * t), sigma(a * t), ([[a * t] - (t - b)] * Maturity / Nostep))
    Next b
    Next a
End If

ReDim DFactor(Nostep) As Double
For a = 1 To Nostep
    DFactor(a) = Discount(Yield(a), (a * dt))
Next a

ReDim sigma(Nostep) As Double

If t = 1 Then 'Dynamically indexed
    For a = 1 To Cashflow
        sigma(a) = Cells(a + 15, 4)
    Next a
Else 'Else Interpolate each step
    For a = 1 To Cashflow
        sigma(a * t) = Cells(a + 15, 4)
    Next a
    For a = 1 To t - 1
        sigma(a * t - (t - b)) = Interpolate([a - 1] * Maturity / Cashflow), [a] * Maturity / Cashflow), sigma((a - 1) * t), sigma(a * t), ([[a * t] - (t - b)] * Maturity / Nostep))
    Next b
    Next a
Else 'Extrapolation
    For a = 1 To Cashflow
        sigma(a) = Cells(a + 15, 4)
    Next a
    Else 'Else Interpolate each step
    For a = 1 To Cashflow
        sigma(a * t) = Cells(a + 15, 4)
    Next a
    For a = 1 To t - 1
        sigma(a * t - (t - b)) = Interpolate([a - 1] * Maturity / Cashflow), [a] * Maturity / Cashflow), sigma((a - 1) * t), sigma(a * t), ([[a * t] - (t - b)] * Maturity / Nostep))
    Next b
    Next a
End If

ReDim Z(Nostep) As Double 'Z(i) in binomial tree, Z = Exp{2 * sigma * (dt ^ 0.5)}

For a = 2 To Nostep
    Z(a) = Exp(2 * sigma(a) * (dt ^ 0.5))
Next a

ReDim f(Nostep + 1, Nostep + 1) As Double

'Newton-Raphson Method
f(1, 1) = Cells(16, 3) 'Initial value
x = 0.1 'Initial value
Count = 0

For t = 2 To Nostep
    y = PV2(t, dt) * DFactor(t)
    Do While Abs(y) > epsilon And Count < 1000
        f(t, 1) = x
        Count = Count + 1
    Loop
Next t
\[ y = PV_2(t, dt) - DFactor(t) \]
\[ x = x - \left( \frac{\partial f(t, dt, DFactor(t))}{\partial x} \right) h \]  Newton-Raphson formula

Count = Count + 1
Loop
If Error < epsilon Then
\[ f(t, 1) = x \]
Else
MsgBox ("No convergent value after 1000 iterations")
End If
Next t
End Sub

' Function Linear interpolation between two known input
Function Interpolate(x1, x2, f1 As Double, f2 As Double, x3)
    dx = x2 - x1
    dFx = f2 - f1
    Interpolate = f1 + ((dFx / dx) * (x3 - x1))
End Function

' Function Discount factor for annual yield given
Function Discount(Yield, Year)
    Discount = 1 / (1 + Yield) ^ Year
End Function

' Function to calculate present value at each node
Function PV1(f1, f2, dt)
    PV1 = 0.5 * ((1 / (1 + (f1 * dt))) + (1 / (1 + (f2 * dt))))
End Function

Function PV2(stage, dt)
    ReDim PV(stage) As Double
    For j = 1 To stage - 1
        f(stage, j + 1) = Z(stage) * f(stage, j)
        PV(j) = PV1(f(stage, j), f(stage, j + 1), dt)
    Next j
    For i = stage - 1 To 2 Step -1
        For j = 1 To i - 1
            f(i, j + 1) = Z(i) * f(i, j)
            PV(j) = (0.5 * PV(j) / (1 + (f(i, j) * dt))) + (0.5 * PV(j + 1) / (1 + (f(i, j + 1) * dt))}
        Next j
        PV(0) = PV(1) / (1 + (f(1, 1) * dt))
    Next i
    PV2 = PV(0)
End Function

Function fh(stage, dt, DFactor)
    Dim h As Double
    h = 0.000001 ' Small increment for derivative
    ReDim fr(stage, stage) As Double
    fr(stage, 1) = f(stage, 1) + h
    ReDim PV(stage) As Double
    For j = 1 To stage - 1
        fr(stage, j + 1) = Z(stage) * fr(stage, j)
        PV(j) = PV1(fr(stage, j), fr(stage, j + 1), dt)
    Next j
    For i = stage - 1 To 2 Step -1
        For j = 1 To i - 1
            f(i, j + 1) = Z(i) * f(i, j)
            PV(j) = (0.5 * PV(j) / (1 + (f(i, j) * dt))) + (0.5 * PV(j + 1) / (1 + (f(i, j + 1) * dt)))
        Next j
        PV(0) = PV(1) / (1 + (f(1, 1) * dt))
        fh = PV(0) - DFactor
    Next i
End Function

' Calculate Present value of bond
' Coupon & Nominal = 100+Coupon
Function Pricebond(dt, Nostep, Cashflow, Maturity, coupon, Frequence, t, Count)
    For a = 1 To Nostep + 1
        Pvbond(Nostep + 1, a) = 100
    Next a
    If Count = 0 Then
        If t = 1 Then
            For i = Nostep To 1 Step -1
                For j = 1 To i - 1
                    f(i, j + 1) = Z(i) * f(i, j)
                    PV(j) = (0.5 * PV(j) / (1 + (f(i, j) * dt))) + (0.5 * PV(j + 1) / (1 + (f(i, j + 1) * dt)))
                Next j
                PV(0) = PV(1) / (1 + (f(1, 1) * dt))
            Next i
            Dim h As Double
            h = 0.000001 ' Small increment for derivative
            ReDim fr(stage, stage) As Double
            fr(stage, 1) = f(stage, 1) + h
            ReDim PV(stage) As Double
            For j = 1 To stage - 1
                fr(stage, j + 1) = Z(stage) * fr(stage, j)
                PV(j) = PV1(fr(stage, j), fr(stage, j + 1), dt)
            Next j
            For i = stage - 1 To 2 Step -1
                For j = 1 To i - 1
                    f(i, j + 1) = Z(i) * f(i, j)
                    PV(j) = (0.5 * PV(j) / (1 + (f(i, j) * dt))) + (0.5 * PV(j + 1) / (1 + (f(i, j + 1) * dt)))
                Next j
                PV(0) = PV(1) / (1 + (f(1, 1) * dt))
                fh = PV(0) - DFactor
            Next i
End Function
Pvbond(i, j) = ((Pvbond(i + 1, j) + coupon) + (Pvbond(i + 1, j + 1) + coupon)) * 0.5 / (1 + (f(i, j) * dt))
Next j
Next i
Else
If t > 1 Then
For i = Nostep To 1 Step -1
If i Mod t = 0 Then 'time coincide with Coupon
For j = 1 To i
Pvbond(i, j) = ((Pvbond(i + 1, j) + i + 1) + coupon) * 0.5 / (1 + (f(i, j) * dt))
Next j
Else
For j = 1 To i
Pvbond(i, j) = (Pvbond(i + 1, j) + coupon) * 0.5 / (1 + (f(i, j) * dt))
Next j
End If
Next i
Else
If t = 1 Then
For i = Nostep To 1 Step -1
For j = 1 To i
Pvbond(i, j) = ((Pvbond(i + 1, j) + coupon) + (Pvbond(i + 1, j + 1) + coupon)) * 0.5 / (1 + (f(i, j) * dt))
Next j
Next i
Else
MsgBox ("Invalid Number of step")
End If
End If
Else
If t = 1 Then
For i = Nostep To 1 Step -1
For j = 1 To i
Pvbond(i, j) = ((Pvbond(i + 1, j) + coupon) + (Pvbond(i + 1, j + 1) + coupon)) * 0.5 / (1 + (f(i, j) * dt))
Next j
Next i
Else
MsgBox ("Invalid Number of step")
End If
End If
End Function
' Function for calculate f(x+h) ; x = spread
Function Fpriceh(dt, Nostep, Cashflow, Maturity, coupon, Frequence, t, Marketprice)
Dim h As Double 'h for calculate Derivative
h = 0.0000001
ReDim fh(Nostep, Nostep) As Double
For a = 1 To Nostep + 1
Pvbond(Nostep + 1, a) = 100
Next a
For a = 1 To Nostep + 1
Pvbond(Nostep + 1, a) = 100
Next a
If t = 1 Then
For i = Nostep To 1 Step -1
For j = 1 To i
Pvbond(i, j) = ((Pvbond(i + 1, j) + coupon) + (Pvbond(i + 1, j + 1) + coupon)) * 0.5 / (1 + (f(i, j) * dt))
Next j
Next i
 Else
MsgBox ("Invalid Number of step")
End If
End Function
End If
End If
Fpriceh = Pvbond(1, 1) - Marketprice
End Function

Sub spread()
Dim Nostep As Integer
Dim Maturity As Double
Worksheets("Variables").Activate
Nostep = Worksheets("Variables").Range("B12").Value
Maturity = Worksheets("Variables").Range("B11").Value 'time to maturity

Dim dt As Double
dt = Maturity / Nostep
Dim Crate As Double
Cr = Cells(9, 2)
Dim Marketprice As Double 'Market Bond price
Marketprice = Cells(10, 5)
Dim epsilon As Double 'Minimum value in Newton-Raphson iterations
epsilon = 0.0000001
Dim Error As Double 'Error in Newton-Raphson iterations
Dim h As Double
h = 0.0000001 'h for calculate derivative
Dim Frequency As Integer
Frequency = Cells(10, 2)
Dim Cashflow As Integer
Cashflow = Frequency * Maturity

t = Nostep / Cashflow ' Number of node between each cash flow
coupon = Crate * 100 / Frequency 'Coupon rate as percent per period
ReDim Pvbond(Nostep + 1, Nostep + 1) As Double
ReDim fs(Nostep, Nostep) As Double 'interest rate plus spread
ReDim spread(Nostep, Nostep) As Double 'interest rate plus spread

Dim spread As Double
spread = Cells(9, 5)
Call OAS 'initial value

If Marketprice > 0 Then
    spread = 0.0001
    Count = 0
    If Marketprice > Pricebond(dt, Nostep, Cashflow, Maturity, coupon, Frequency, t, Count) Then
        spread = 0
        Else
        y = Pricebond(dt, Nostep, Cashflow, Maturity, coupon, Frequency, t, Count) - Marketprice 'Function f(x)= Bondprice-Marketprice
        Do While Abs(y) > epsilon And Count < 10000 'Newton-Raphson iterations
            Count = Count + 1
            For i = 1 To Nostep 'input spread to function
                For j = 1 To i
                    fs(i, j) = f(i, j) + spread
                    Next j
                Next i
            y = Pricebond(dt, Nostep, Cashflow, Maturity, coupon, Frequency, t, Count) - Marketprice
            spread = spread - (y / ((Fpriceh(dt, Nostep, Cashflow, Maturity, coupon, Frequency, t, Marketprice) - y) / h))
        Loop
        If Abs(y) < epsilon Then
            Else
                MsgBox ("No convergence value after 10000 iterations.")
        End If
    End If
Else
    MsgBox ("Input Market Bond Price")
End If

End Sub
Sub Spreadtomarket()
Dim Nostep As Integer
Dim Maturity As Double
Worksheets("Variables").Activate
Nostep = Worksheets("Variables").Range("B12").Value
Maturity = Worksheets("Variables").Range("B11").Value 'time to maturity

Dim dt As Double
dt = Maturity / Nostep
Dim Crate As Double
Cr = Cells(9, 2)
Dim Marketprice As Double 'Market Bond price
Marketprice = Cells(10, 5)
Dim Frequency As Integer
Frequency = Cells(10, 2)
Dim Cashflow As Integer
Cashflow = Frequency * Maturity

t = Nostep / Cashflow ' Number of node between each cash flow
coupon = Crate * 100 / Frequency 'Coupon rate as percent per period
ReDim Pvbond(Nostep + 1, Nostep + 1) As Double
ReDim fs(Nostep, Nostep) As Double 'interest rate plus spread
ReDim spread(Nostep, Nostep) As Double 'interest rate plus spread

Dim spread As Double
spread = Cells(9, 5)
If spread > 0 Then
Call OAS
For i = 1 To Nostep 'input spread to function
 For j = 1 To i
 f(i, j) = f(i, j) + spread
 Next j
Next i
Marketprice = Pricebond(dt, Nostep, Cashflow, Maturity, coupon, Frequence, t, 1)
Cells(10, 5) = Marketprice
Else
MsgBox("Input Spread")
End If
End Sub
Sub Trees()
Call OAS
Dim Nostep As Integer
Dim Maturity As Double
Dim spread As Double
Nostep = Worksheets("Variables").Range("B12").Value
Maturity = Worksheets("Variables").Range("B11").Value
dt = Maturity / Nostep
Worksheets("Tree").Activate
Cells.Delete
'Rate tree
Cells(2, 1).Value = "Rate tree"
Cells(2, 1).Select
Selection.Font.Bold = True
For i = 1 To Nostep
 For j = 1 To 2 * i Step 2
 Cells(Nostep + j - i + 2, i + 1).Value = f(i, (j + 1) / 2)
 Cells(Nostep + j - i + 2, i + 1).Select
 Selection.Style = "Output"
 Selection.NumberFormat = "0.0000"
 Next j
Next i
'Rate tree using the spread
Cells(Nostep * 2 + 2, 1).Value = "Rate tree using the spread"
Cells(Nostep * 2 + 2, 1).Select
Selection.Font.Bold = True
spread = Worksheets("Variables").Range("E9").Value
For i = 1 To Nostep
 For j = 1 To 2 * i Step 2
 Cells(Nostep * 3 + j - i + 2, i + 1).Value = f(i, (j + 1) / 2) + spread
 Cells(Nostep * 3 + j - i + 2, i + 1).Select
 Selection.Style = "Output"
 Selection.NumberFormat = "0.0000"
 Next j
Next i
'Discount factor tree using the spread
Cells(Nostep * 4 + 2, 1).Value = "Discount factor tree using the spread"
Cells(Nostep * 4 + 2, 1).Select
Selection.Font.Bold = True
spread = Worksheets("Variables").Range("E9").Value
For i = 1 To Nostep + 1
 f(Nostep + 1, i) = 100
Next i
spread = Worksheets("Variables").Range("E9").Value
For i = 1 To Nostep + 1
 f(Nostep + 1, i) = 100
Next i
For j = 1 To 2 * (Nostep + 1) Step 2
Cells(Nostep * 7 + j - (Nostep + 1) + 3, (Nostep + 1) + 1).Value = f((Nostep + 1), (j + 1) / 2)
Cells(Nostep * 7 + j - (Nostep + 1) + 3, (Nostep + 1) + 1).Select
Selection.Style = "Output"
Selection.NumberFormat = "0.0000"
Next j

For i = Nostep To 1 Step -1
For j = 2 * i To 1 Step -2
Cells(Nostep * 7 + j - i + 2, i + 1).Value = (Cells(Nostep * 7 + j - i + 2) - 1, i + 1) + 1.Value + Cells(Nostep * 7 + j - i + 2) + 1, i + 1) + 1.Value + Cells(Nostep * 7 + j - i + 2) + 1, i + 1).Select
Selection.Style = "Output"
Selection.NumberFormat = "0.0000"
Next j
Next i

'Coupon paying bond price using the spread
Cells(Nostep * 8 + 4, 1).Value = "Coupon paying bond price using the spread"
Cells(Nostep * 8 + 4, 1).Select
Selection.Font.Bold = True
spread = Worksheets("Variables").Range("E9").Value
Arate = Worksheets("Variables").Range("B9").Value 'Annual coupon rate
Frequency = Worksheets("Variables").Range("B10").Value 'Coupons per year
Times = Worksheets("Variables").Range("B13").Value 'Times of cash flow
Crate = Arate / Frequency 'Coupon rate

For i = 1 To Nostep + 1
f(Nostep + 1, i) = 100 + (Crate * 100)
Next i
For j = 1 To 2 * (Nostep + 1) Step 2
Cells(Nostep * 9 + j - (Nostep + 1) + 5, (Nostep + 1) + 1).Value = f((Nostep + 1), (j + 1) / 2)
Cells(Nostep * 9 + j - (Nostep + 1) + 5, (Nostep + 1) + 1).Select
Selection.Style = "Output"
Selection.NumberFormat = "0.0000"
Next j

For i = Nostep To 1 Step -1
For j = 2 * i To 1 Step -2
if (Nostep + 1) - i = Times Then
if i > 1 Then
Cells(Nostep * 9 + j - i + 4, i + 1).Value = Cells(Nostep * 9 + j - i + 4, i + 1).Value + (Crate * 100)
End If
End if
Cells(Nostep * 9 + j - i + 4, i + 1).Select
Selection.NumberFormat = "0.0000"
Next j
Next i

'Callable coupon paying bond price using the spread
Cells(Nostep * 10 + 6, 1).Value = "Callable coupon paying bond price using the spread"
Cells(Nostep * 10 + 6, 1).Select
Selection.Font.Bold = True
spread = Worksheets("Variables").Range("E9").Value
Arate = Worksheets("Variables").Range("B9").Value 'Annual coupon rate
Frequency = Worksheets("Variables").Range("B10").Value 'Coupons per year
Times = Worksheets("Variables").Range("B13").Value 'Times of cash flow
Crate = Arate / Frequency 'Coupon rate
Cashflow = Frequency * Maturity 'Number of cash flow

'Strike vector
ReDim Strike(Cashflow, Times) As Double
ReDim StrikeStep(Nostep + Times + 1) As Single
a = Times + 1
For i = 1 To Cashflow - 1
if Worksheets("Variables").Cells(i + 15, 6).Value = "Period" Then
For j = 1 To Times
Strike(i, j) = Worksheets("Variables").Cells(i + 15, 5).Value
StrikeStep(a) = Strike(i, j)
a = a + 1
Next j
Else
    Strike[i, 1] = Worksheets("Variables").Cells(i + 15, 5).Value
    StrikeStep(a) = Strike[i, 1]
    a = a + 1
For j = 2 To Times
    Strike[i, j] = 0
    StrikeStep(a) = Strike[i, j]
    a = a + 1
Next j
End If
Next i

For i = 1 To Nostep + 1
    f(Nostep + 1, i) = 100 + (Crate * 100)
Next i

For j = 1 To 2 * (Nostep + 1) Step 2
    Cells(Nostep * 11 + j - (Nostep + 1) + 7, (Nostep + 1) + 1).Value = f((Nostep + 1), (j + 1) / 2)
    Cells(Nostep * 11 + j - (Nostep + 1) + 7, (Nostep + 1) + 1).Select
    Selection.Style = "Output"
    Selection.NumberFormat = "0.0000"
Next j

For i = Nostep To 1 Step -1
    For j = 2 * i To 1 Step -2
        Cells(Nostep * 11 + j - i + 6, i + 1).Value = (Cells((Nostep * 11 + j - i + 6) - 1, (i + 1) + 1).Value + Cells((Nostep * 11 + j - i + 6) + 1, (i + 1) + 1).Value) * 0.5 * (1 / (1 + (f(i, j / 2) + spread) * dt))
        If StrikeStep(i) > 0 Then
            If Cells(Nostep * 11 + j - i + 6, i + 1).Value > StrikeStep(i) Then
                Cells(Nostep * 11 + j - i + 6, i + 1).Value = StrikeStep(i)
            End If
        End If
        Cells(Nostep * 11 + j - i + 6, i + 1).Select
        Selection.Style = "Output"
        Selection.NumberFormat = "0.0000"
    End If
Next j

If i = 1 Then
    b2 = Cells(Nostep * 11 + j - i + 6, i + 1)
End If
Next j

If StrikeStep(i) > 0 Then
    If Cells(Nostep * 11 + j - i + 6, i + 1).Value > StrikeStep(i) Then
        Cells(Nostep * 11 + j - i + 6, i + 1).Value = StrikeStep(i)
    End If
    Cells(Nostep * 11 + j - i + 6, i + 1).Select
    Selection.Style = "Output"
    Selection.NumberFormat = "0.0000"
    If i = 1 Then
        b2 = Cells(Nostep * 11 + j - i + 6, i + 1)
    End If
    Next j
End If

Worksheets("Variables").Cells(12, 5).Value = b1 - b2
Worksheets("Variables").Cells(11, 5).Value = b1

End Sub

Sub OptionPrice()
    Call OAS
    Dim Nostep As Integer
    Dim Maturity As Double
    Dim spread As Double
    Nostep = Worksheets("Variables").Range("B12").Value
    Maturity = Worksheets("Variables").Range("B11").Value
    dt = Maturity / Nostep
    'Coupon paying bond price using the spread
    ReDim Bullet(Nostep * 3, Nostep * 3) As Double
    spread = Worksheets("Variables").Range("E9").Value
    Arate = Worksheets("Variables").Range("B9").Value 'Annual coupon rate
    Frequence = Worksheets("Variables").Range("B10").Value 'Coupons per year
    Times = Worksheets("Variables").Range("B13").Value 'Times of cash flow
    Crate = Arate / Frequence 'Coupon rate
    For i = 1 To Nostep + 1
        f(Nostep + 1, i) = 100 + (Crate * 100)
    Next i
    For j = 1 To 2 * (Nostep + 1) Step 2
        Bullet(Nostep + j - (Nostep + 1) + 1, (Nostep + 1) + 1) = f((Nostep + 1), (j + 1) / 2)
    Next j
For i = Nostep To 1 Step -1
    For j = 2 * i To 1 Step -2
        Bullet(Nostep + j - i + 0, i + 1) = (Bullet(Nostep + j - i + 0) - 1, (i + 1) + 1) + Bullet((Nostep + j - i + 0) + 1, (i + 1) + 1)) * 0.5 * (1 / (1 + (f(i, j / 2) + spread) * dt))
    Next j
Next i

If (Nostep + 1) - i = Times Then
    If i > 1 Then
        Bullet(Nostep + j - i + 0, i + 1) = Bullet(Nostep + j - i + 0, i + 1) + (Crate * 100)
    End If
End If

If i = 1 Then
    b1 = Bullet(Nostep + j - i + 0, i + 1)
End If

Next j

If (Nostep + 1) - i = Times Then
    Times = 2 * Times
End If

Next i

'Callable coupon paying bond price using the spread

ReDim Callable(Nostep * 3, Nostep * 3) As Double

spread = Worksheets("Variables").Range("E9").Value
Arate = Worksheets("Variables").Range("B9").Value 'Annual coupon rate
Frequency = Worksheets("Variables").Range("B10").Value 'Coupons per year
Times = Worksheets("Variables").Range("B13").Value 'Times of cash flow
Crates = Arate / Frequency 'Coupon rate
Cashflows = Frequency * Maturity 'Number of cash flow

'Strike vector

ReDim Strike(Cashflow, Times) As Double
ReDim StrikeStep(Nostep + Times + 1) As Double

For i = 1 To Cashflow - 1
    If Worksheets("Variables").Cells(i + 15, 6).Value = "Period" Then
        For j = 1 To Times
            Strike(i, j) = Worksheets("Variables").Cells(i + 15, j).Value
            StrikeStep(a) = Strike(i, j)
            a = a + 1
        Next j
    Else
        Strike(i, 1) = Worksheets("Variables").Cells(i + 15, 5).Value
        StrikeStep(a) = Strike(i, 1)
        a = a + 1
        For j = 2 To Times
            Strike(i, j) = 0
            StrikeStep(a) = Strike(i, j)
            a = a + 1
        Next j
    End If
Next i

For i = 1 To Nostep = 1
    f(Nostep + 1, i) = 100 + (Crate * 100)
Next i

For j = 1 To 2 * (Nostep + 1) Step 2
    Callable(Nostep + j - (Nostep + 1) + 1, (Nostep + 1) + 1) = f(Nostep + 1, (j + 1) / 2)
Next j

For i = 1 To Nostep To 1 Step -1
    For j = 2 * i To 1 Step -2
        Callable(Nostep + j - i + 0, i + 1) = Callable((Nostep + j - i + 0) - 1, (i + 1) + 1) + Callable((Nostep + j - i + 0) + 1, (i + 1) + 1)) * 0.5 * (1 / (1 + (f(i, j / 2) + spread) * dt))
    Next j
Next i

If (Nostep + 1) - i = Times Then
    If i > 1 Then
        Callable(Nostep + j - i + 0, i + 1) = Callable(Nostep + j - i + 0, i + 1) + (Crate * 100)
    End If
End If

If StrikeStep(i) > 0 Then
    If Callable(Nostep + j - i + 0, i + 1) > StrikeStep(i) Then
        Callable(Nostep + j - i + 0, i + 1) = StrikeStep(i)
    End If
End If

If i = 1 Then
    b2 = Callable(Nostep + j - i + 0, i + 1)
End If

Next j

If (Nostep + 1) - i = Times Then
    Times = 2 * Times
End If
End If
Next i

Worksheets("Variables").Cells(12, 5).Value = b1 - b2
Worksheets("Variables").Cells(11, 5).Value = b1

End Sub

Function Convergence(Nostep, Maturity, spread, Arate, Frequence, Times, xx)

dt = Maturity / Nostep

'Coupon paying bond price using the spread
ReDim Bullet(Nostep * 3, Nostep * 3) As Double

temp_1 = Times
Crate = Arate / Frequence 'Coupon rate

For i = 1 To Nostep + 1
f(Nostep + 1, i) = 100 + (Crate * 100)
Next i

For j = 1 To 2 * (Nostep + 1) Step 2
Bullet(Nostep + j - (Nostep + 1) + 1, (Nostep + 1) + 1) = f(Nostep + 1, (j + 1) / 2)
Next j

For i = Nostep To 1 Step -1
For j = 2 * i To 1 Step -2
Bullet(Nostep + j - i + 0, i + 1) = (Bullet((Nostep + j - i + 0) - 1, (i + 1) + 1) + Bullet((Nostep + j - i + 0) + 1, (i + 1) + 1)) * 0.5 * (1 / (1 + (f(i, j / 2) + spread) * dt))
If (Nostep + 1) - i = temp_t Then
If i > 1 Then
Bullet(Nostep + j - i + 0, i + 1) = Bullet((Nostep + j - i + 0) + 1, (i + 1) + 1) + (Crate * 100)
End If
End If

If i = 1 Then
b1 = Bullet(Nostep + j - i + 0, i + 1)
End If
Next j

If (Nostep + 1) - i = temp_t Then
temp_t = 2 * temp_t
End If
Next i

'Callable coupon paying bond price using the spread
ReDim Callable(Nostep * 3, Nostep * 3) As Double

Cashflow = Frequence * Maturity 'Number of cash flow

temp_t = Times
Strike vector
ReDim Strike(Cashflow, Times) As Double
ReDim StrikeStep(100) As Single

a = temp_t + 1

For i = 1 To Cashflow - 1
If Worksheets("Variables").Cells(i + 15, 6).Value = "Period" Then
For j = 1 To temp_t
Strike(i, j) = Worksheets("Variables").Cells(i + 15, 5).Value
StrikeStep(a) = Strike(i, j)
a = a + 1
Next j
Else
Strike(i, 1) = Worksheets("Variables").Cells(i + 15, 5).Value
StrikeStep(a) = Strike(i, 1)
a = a + 1
For j = 2 To temp_t
Strike(i, j) = 0
StrikeStep(a) = Strike(i, j)
a = a + 1
Next j
End If
Next i

For i = 1 To Nostep + 1
f(Nostep + 1, i) = 100 + (Crate * 100)
Next i

For j = 1 To 2 * (Nostep + 1) Step 2
Callable(Nostep + j - (Nostep + 1) + 1, (Nostep + 1) + 1) = f(Nostep + 1, (j + 1) / 2)
Next j

For i = Nostep To 1 Step -1
For j = 2 * i To 1 Step -2
Callable(Nostep + j - i + 0, i + 1) = (Callable((Nostep + j - i + 0) - 1, (i + 1) + 1) + Callable((Nostep + j - i + 0) + 1, (i + 1) + 1)) * 0.5 * (1 / (1 + (f(i, j / 2) + spread) * dt))
If (Nostep + 1) - i = temp_t Then
If i > 1 Then
Callable(Nostep + j - i + 0, i + 1) = Callable(Nostep + j - i + 0, i + 1) + (Crate * 100)
End If
End If

If StrikeStep(i) > 0 Then
If Callable(Nostep + j - i + 0, i + 1) > StrikeStep(i) Then
Callable(Nostep + j - i + 0, i + 1) = StrikeStep(i)
End If
End If

If i = 1 Then
b2 = Callable(Nostep + j - i + 0, i + 1)
End If
Next j

If (Nostep + 1) - i = Times Then
temp_t = 2 * temp_t
End If
Next i

Convergence = b1 - b2
Worksheets("Convergence").Cells(xx + 1, 3) = b1
End Function

Sub Convert()
    Worksheets("Convergence").Activate
    Cells.Delete
    Worksheets("Convergence").Cells(1, 1) = "Times of Cash flow"
    Worksheets("Convergence").Cells(1, 2) = "Option value"
    Worksheets("Convergence").Cells(1, 3) = "Bullet bond"

    Worksheets("Variables").Cells(13, 2) = 1
    Call OptionPrice
    Worksheets("Convergence").Cells(1 + 1, 1) = 1
    Worksheets("Convergence").Cells(1 + 1, 2) = Worksheets("Variables").Cells(12, 5)
    Worksheets("Convergence").Cells(1 + 1, 3) = Worksheets("Variables").Cells(11, 5)

    For i = 1 To 6
        x = 2 ^ i
        Worksheets("Variables").Cells(13, 2) = x
        Call OptionPrice
        Worksheets("Convergence").Cells(i + 2, 2) = Worksheets("Variables").Cells(12, 5)
        Worksheets("Convergence").Cells(i + 2, 3) = Worksheets("Variables").Cells(11, 5)
    Next i

    'Build chart
    Range("A2:B8").Select
    Worksheets("Convergence").Select
    ActiveSheet.Shapes.AddChart(30, 250, 100, 400, 250).Select
    ActiveChart.SetSourceData Source:=Range("Convergence!$A$1:$B$8")
    ActiveChart.ChartType = xlXYScatterSmooth
    ActiveSheet.ChartObjects(1).Activate
    ActiveChart.Axes(xlValue).MajorGridlines.Select
    ActiveChart.SeriesCollection(1).Delete
    ActiveChart.SeriesCollection(1).XValues = "+Convergence!$A$2:$A$8"

    'Apply Chart Layout
    ActiveChart.ApplyLayout (1)
    ActiveChart.ChartTitle.Select
    ActiveChart.ChartTitle.Text = "Convergence"
    ActiveChart.Axes(xlCategory, xlPrimary).AxisTitle.Text = "Time of Cash Flows"
    ActiveChart.Axes(xlValue, xlPrimary).AxisTitle.Text = "Option Price"

    Range("A2:A7,C2:C7").Select
    ActiveSheet.Shapes.AddChart(30, 600, 100, 400, 250).Select
    ActiveChart.SetSourceData Source:=Range("Convergence!$A$1:$A$8,$C$1:$C$8")
    ActiveChart.ChartType = xlXYScatterSmooth
    ActiveSheet.ChartObjects(2).Activate
    ActiveChart.Axes(xlValue).MajorGridlines.Select
    ActiveChart.SeriesCollection(1).Delete
    ActiveChart.SeriesCollection(1).XValues = "+Convergence!$A$2:$A$8"
    ActiveChart.SeriesCollection(1).Select
    ActiveChart.ChartStyle = 2

    'Apply Chart Layout
    ActiveChart.ApplyLayout (1)
    ActiveChart.ChartTitle.Select
    ActiveChart.ChartTitle.Text = "Convergence"
    ActiveChart.Axes(xlCategory, xlPrimary).AxisTitle.Text = "Time of Cash Flows"
    ActiveChart.Axes(xlValue, xlPrimary).AxisTitle.Text = "Bullet Bond Price"

End Sub